Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

Mapping and Visualizing Ancient Water Storage Systems with an ROV – An Approach Based on Fusing Stationary Scans within a Particle Filter

William McVicker, Jeffrey Forrester, Timothy Gambin, Jane Lehr, Zoë J. Wood, and Christopher M. Clark

***Abstract*— This paper presents a new method for constructing 2D maps of enclosed underwater structures using an underwater robot equipped with only a 2D scanning sonar, compass and depth sensor. In particular, no motion model or odometry is used. To accomplish this, a two step offline SLAM method is applied to a set of stationary sonar scans. In the first step, the change in position of the robot between each consecutive pair of stationary sonar scans is estimated using a particle filter. This set of pair wise relative scan positions is used to create an estimate of each scan's position within a global coordinate frame using a weighted least squares fit that optimizes consistency between the relative positions of the entire set of scans. In the second step of the method, scans and their estimated positions act as inputs to a mapping algorithm that constructs 2D octree-based evidence grid maps of the site.**

This work is motivated by a multi-year archeological project that aims to construct maps of ancient water storage systems, i.e. cisterns, on the islands of Malta and Gozo. Cisterns, wells, and water galleries within fortresses, churches and homes operated as water storage systems as far back as 2000 B.C. Using a Remotely Operated Vehicle (ROV) these water storage systems located around the islands were explored while collecting video, still images, sonar, depth, and compass measurements. Data gathered from 3 different expeditions has produced maps of over 60 sites. Presented are results from applying the new mapping method to both a swimming pool of known size and to several of the previously unexplored water storage systems.

I. Introduction

Underwater robots are used to explore harsh environments, dangerous caves, and underwater domains. Remotely Operated Vehicles (ROV) allow researchers to safely study these places remotely by capturing video, images, acoustic data, and measurements from underwater sensors. The field of underwater robotics in the past decade has made substantial progress in the areas of localization and mapping. Sonar technology has been the primary choice of equipment to assist in mapping underwater environments because sonar waves propagate through water better than light. Recently in [1], an ROV was used to inspect 1 km of an underwater tunnel operated by Electricité de France for reasons of availability, safety, accessibility, and diagnostic quality. Another example of underwater ROV research includes implementing vision systems for underwater applications, to support shared control, and 3D mapping [2].

Fig. 1: The VideoRay Pro 3 GTO is an underwater micro-ROV with dimensions 36.8cm x 28.9cm x 21.6cm. It is equipped with depth and compass sensors along with a front and a rear video camera. Additionally, a Tritech Micron scanning sonar can be attached to the ROV.

More relevant, micro-sized ROVs have been developed to help improve maneuverability within tight passages for a variety of applications including the exploration of sensitive ecosystems that one may not want to disrupt [3]. Other environments ideal for micro-ROVs include ancient cisterns, wells, and water galleries. The water storage systems found in Malta are difficult to access due to their size (e.g. a typical opening diameter of 0.3 m). To explore such envrionments, a VideoRay Pro 3 ROV equipped with an underwater scanning sonar head, depth sensor, and two video cameras, (seen in Fig. 1), was used to generate 2D and 3D sonar based maps. This paper proposes a new mapping and localization technique used to reconstruct the explored cisterns as well as document results from a recent expedition, (i.e. Spring of 2012).

Since the sites explored in this research have tunnels of limited size and accessibility, the ROV sensor payload must be minimal. Much research has focussed on developing complex motion models to decrease uncertainty, accurately model the ROV's motion, and improve localization. Instead of increasing the sensor payload of the ROV, this paper proposes a 2D localization algorithm to be run offline that does

This material is based upon work supported by the National Science Foundation under Grant No. 0966608.

W. McVicker and J. Forrester are Computer Science students while J. Lehr and Z. J. Wood are professors of Ethnic Studies and Computer Science respectively at California Polytechnic State University, San Luis Obispo, CA 93407, USA wmcvicke@, jdforres@, jlehr@, zwood@calpoly.edu

T. Gambin is a professor of Archaeology at the University of Malta, Msida MSD 2080, Malta and a member of the AURORA Trust tgambin@hotmail.com

C. M. Clark is a professor in the Dept. of Engineering at Harvey Mudd College 301 Platt Boulevard, Claremont, CA 91711, USA clark@hmc.edu

978-1-4673-2127-3/12/$31.00 © 2012 IEEE

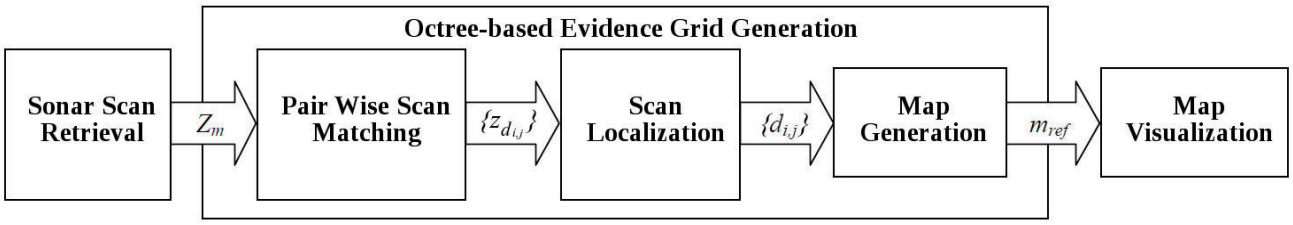


Fig. 2: The pipeline created to generate 3D maps

not require a motion model, but instead uses a particle filter to calculate relative translations between pairs of sonar scans that are used to construct a global map of the environment.

This paper is sectioned into five parts. In Section II, a brief background is provided on similar mapping and localization techniques. Section III explains the mapping reconstruction pipeline used to generate a map. In section IV, the results are presented followed by section V with the current conclusions and proposed future work.

II. Background

The recent advances of underwater robot sensing technology (e.g. sonar, imaging, doppler velocity logging) have led to the ability to conduct Simultaneous Localization and Mapping (SLAM) [4] in the underwater domain [5]. In many SLAM algorithms, it is common to use robot odometry to predict the new robot position with respect to the map before using exteroceptive sensor measurements to correct the robot's position and update the map itself [6].

Analyzing raw measurements, as opposed to extracting geometric features from a map, is known as scan matching. A probabilistic algorithm called iterative-closest-points (ICP) has taken popularity among the robotics community in many different variations [7], [8], and [9]. In [10], the sonar probabilistic model spIC was used to localize a mobile robot by analyzing raw sonar data to correct odemetry errors for short robot trajectories. This helped minimize the displacement between noisy and sparse measurements.

Such applications have proven to work well in underwater environments. The core of the ICP algorithm matches two point clouds together in order to align the scans for map generation and/or vehicle localization. This algorithm has proven to work well with noisy data, but remains computationally heavy – $O(P^3)$ per iteration for P number of points. In [9], Fairfield and Wettergreen developed a variation to ICP called icLK to generate 3D maps of underground mines which reduced the compexity of ICP to $O(P^2)$ by thresholding the data; however, with 100 k points the approach remains burdensome.

Other recent research has focused on generating maps in real-time using a complex motion model in order to associate each sonar measurement with a corresponding location in the map. For example in [8], a pose-based algorithm was developed to map unstructured and unfamiliar environments using a probabilistic scan matching technique. The scan matching techniques that extract ranges from sonar beans explained in [10], [11], and [8] are most similar to the techniques developed in this paper.

There has also been an increase in visual SLAM recently. A technique called frameSLAM [12] uses bundle adjustment techniques to match point features along with stereo vision to track landmarks. The system developed was capable of autonomously navigating an offroad vehicle with only the use of stero vision. Visual SLAM has proven to work well in terrestial environments, but murkey water (a common condition in cisterns) would likely decrease performance.

Similar to detecting frames and features, the incremental smoothing and mapping (iSAM) [13] technique uses an informational filter to incrementally associate measurements in large-scale environments to solve the full SLAM problem. iSAM and frameSLAM are both feature based which differs from the scan matching solution based on raw measurements presented in this paper.

In the previous expeditions dedicated to mapping Maltese cisterns [14], [15], [16] the mapping techniques included sonar mosaicking, and underwater robot SLAM with both a stationary and moving robot. Sonar mosaics are images generated by piecing together different parts (scans) of the image to create a single image. This is a manual and time consuming job, but is able to successfully localized an ROV through manual calculations. For underwater robot SLAM, inadequate motion modeling led to reduced accuracy in robot localization and hence mapping [15]. To ensure highly accurate maps, the subsequent expeditions focussed on obtaining a series of stationary sonar scans from several positions in the tunnel [14].

The work reported here differs from the scan matching techniques developed in [8] and [11] in that our robot has a limited payload and no motion model or odometry is used to predict the robot's location. Instead, mapping is done offline. A particle filter is first used to calculate relative positions of the robot between consecutive pairs of stationary scans. Then, a weighted least squares approach uses these relative positions to calculate the absolute position of the robot for each individual sonar scan. To note, the algorithm improves the consistency of scan matching by considering how every tuple and every triplet of scans fit together. Finally, the scans and their estimated positions are used to create a 2D octree-based evidence grid map. A detailed description of this approach is presented below.

III. Mapping Reconstruction Pipeline

In [15], it proved difficult to construct maps using sonar data collected while moving. The localization uncertainty accumulated at a far higher rate than could be corrected with infrequent sonar measurements. This work attempts to accomplish the following: given a series of stationary sonar scans with corresponding depth and compass measurements, where each scan overlaps with at least the immediately following scan in the series, determine the locations of the ROV scans such that SLAM can be accomplished with relative scan positions and scan data.

The proposed solution uses the following stages to generate a map: Data Collection, Pair Wise Scan Matching, Scan Localization, Mapping, and Visualization. These stages are shown in Fig. 2.

A. Data Collection

Data collection was performed using the VideoRay Pro 3 GTO as described in Sec. I. The process of collecting data begins by visually investigating the site and sketching the surroundings in order for one to uniquely identify each site from a top-down perspective. The ROV is then lowered into the cistern for exploration. Data is collected as follows: land the ROV on the bottom or hover mid-depth without moving, log ROV depth and compass sensor measurements, initiate a sonar scan for one full revolution, move forward and repeat.

The collection of these measurements are defined as Z_m and collected at each time step t such that $Z_m = \{Z_m^t | t = 0...T_{max}\}$ where Z_m^t represents all the sensor measurements collected at time t, i.e at a new scan location. These include robot yaw angle measurements z_θ, depth measurement z_d, and stationary sonar scans z_s.

$$Z_m^t = [z_\theta \; z_d \; z_s]^t \tag{1}$$

Each sonar scan z_s consists of a series of $j = 1..A$ scan angles α_j, each with a corresponding vector of signal strengths $[ss_{j,i}]$. These signal strengths represent the echo intensities of the discritized sonar signal returned from a specific distance normal to the sensor that increases linearly with the value of $i = 1..Num\_Bins$ where Num_Bins is defined when configuring the sonar head.

$$z_s = \{[\alpha_j \; ss_{j,1} \; ... \; ss_{j,Num\_Bins}] \mid j = 1..A\} \tag{2}$$

For cisterns that are bell-shaped or organically structured, scans must be taken at multiple depths.

B. Pair Wise Scan Matching

Pair Wise Scan Matching takes the sensor measurements Z_m as input and outputs the measured position translations $z_{d_{ij\_m}}$ of the robot between each pair of stationary scans i and j. To note, these relative translation vectors are aligned with a global coordinate frame that has the X-axis aligned with the direction of true North.

The pseudo code for Pair Wise Scan Matching is shown in Table I. To begin, the robot's position during scan s_i is assumed to be at the origin of a cartesian coordinate frame and map m_i is generated with an octree-based evidence grid (line 2) within this coordinate frame using a log-likelihood approach [17].

TABLE I: Pair Wise Scan Matching Algorithm

	Calc_Translation_Measurements(*s, Num_Scans*)
1:	*for i = 1 to Num_Scans do*
2:	m_i = **construct_map**(s_i)
3:	*for j = i-1 to i+1 do*
4:	d_{ij_m} = **PF_Localization**(m_i, s_j)
5:	*endfor*
6:	*endfor*

TABLE II: Particle Filter Robot Localization

	PF_Localization(m_i, s_j)
1:	**initialize_particle_states**()
2:	*for i = 1 to Num_Iterations do*
3:	*for k = 1 to Num_Particles do*
4:	X_k = **propagate_robot_state**(X_k)
5:	w_{ij}^k = **calculate_weight**(m_i, s_j)
6:	*endfor*
7:	**resample_particles**()
8:	*if* **particles_converged**() *break*
9:	*endfor*
10:	d_{ij} = **calculate_translation**(d_m, w_μ)

In generating the map, the input data requires filtering due to the noisy characteristics of the sonar data. The raw sonar echo intensities negatively affect the scan matching algorithm introducing inaccurate alignments caused by mistaking noise as a wall. To handle this, a similar technique to beam segmentation found in [11] was developed.

First, the signal strengths are normalized between 0 and 255 based on the maximum echo intesity found in the entire scan. The echo intensities are then converted into range measurements by iterating through each beam of the scan and identifying the bin with the maximum echo intensity that has at least one adjacent bin along the same beam. If the maximum intensity is above a set threshold (50 in our case), the bin's value and the adjacent bins' values are set to 255 while all the other bins in the beam are set to 0. This approach removes most of the noise and allows for two scans with drastically different levels of echo intensities to be accurately aligned.

Next, on line 4, the location of the robot during scan j with respect to map m_i is calculated using an implementation of Particle Filter Localization [17] .

In this Particle Filter, as described in Table II, a collection of $k = 1..Num\_Particles$ particles is used to represent the robot state during scan j. Each particle k consists of the robot's state $[x^k y^k z^k \theta^k]$, and a weight w^k that indicates how likely particle k represents the true state. The particle position states are initially sampled randomly (line 1 of Table II) from a square uniform distribution centered on the origin of m_i and with dimensions $L_{init} \times L_{init}$ meters.

The Particle Filter iterates for $Num\_Iterations$ or until the particles converge. At each iteration of the algorithm, the x, y, and θ of the particle's state are propagated (line 4) by adding a sample drawn from a zero mean gaussian distribution of variance σ_p^2. This added randomness models

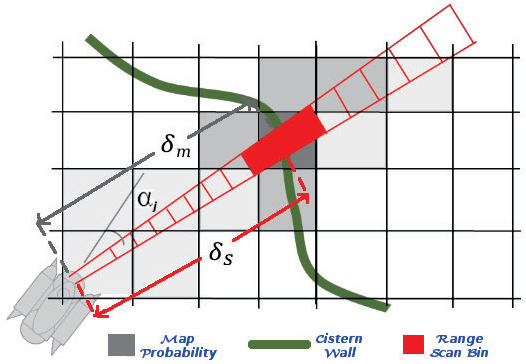


Fig. 3: The particles' weights are calculated by taking the difference between the distance to a wall in the robot's map and in the range data with respect to the robot

errors introduced by drift, sensor measurements, and driver control.

To calculate particle weights (line 5), scan s_j's range data is compared to the map m_i . Particle k's weight w_{ij}^k is calculated as shown in (4) by calculating the difference between δ_s, the distance to a wall acccording to the scan s_j's range measurement scan angle α_j, and δ_m calculated as the distance from particle k's robot state in the direction of α_j to the nearest occupied cell with a probability above *WEIGHT_THRESHOLD* in map m_i as demonstrated in Fig. 3.

This difference $\delta_s - \delta_m$ is plugged into the Gaussian function $\varphi(\delta_s, \delta_m)$ (3) where the standard deviation σ_c equals twice the map's cell size r_c with parameter a set to equate each of the Gaussain Function's maximum value to one. If this difference is less than a set maximum distance, δ_{max}, then the particle's weight w_{ij}^k is equal to $\varphi(\delta_s, \delta_m)$; otherwise, w_{ij}^k is reduced by $1 - \varphi(\delta_s, \delta_m)$ (4) in order to include negative feedback.

$$\varphi(\delta_s, \delta_m) = \frac{1}{a\sqrt{2\pi}} e^{-\frac{(\delta_s - \delta_m)^2}{2\sigma_c^2}} \tag{3}$$

$$w_{ij}^k = \begin{cases} \varphi(\delta_s, \delta_m) & if\ \delta_s - \delta_m < \delta_{max} \\ w_{ij}^k - (1 - \varphi(\delta_s, \delta_m)) & otherwise \end{cases} \tag{4}$$

For every iteration, the algorithm creates a new set of particles (line 7) by resampling from the current set of particles randomly where each particle has a probability of being selected proporational to its weight w_{ij}^k.

In order to determine if the particles have converged (line 8), an exponential average of the particles' weights $w_{ij\_\mu}^t = \chi_w w_{ij}^t + (1 - \chi_w) w_{ij\_\mu}^{t-1}$ and an exponential average of the standard deviation of the particles' states $\sigma_{p\_\mu}^t = \chi_\sigma \sigma_p^t + (1 - \chi_\sigma) \sigma_{p\_\mu}^{t-1}$ are calculated before the particles are resampled where χ_w and $\chi_\sigma \in [0, 1]$ are constant smoothing factors. The algorithm iterates for scans i and j until the particles converge or *Num_Iterations* is exceeded. Convergence is determined when both of the following criteria are met:

- The exponential average of the standard deviation of the particles' states $\sigma_{p\_\mu}^t$ is less than or equal to the robot's map cell size r_c.

$$\sigma_{p\_\mu}^t \leq r_c \tag{5}$$

- The exponential average of the particles' weights $w_{ij\_\mu}^t$ is greater than or equal to an experimentally determined threshold τ_{pw}.

$$w_{ij\_\mu}^t \geq \tau_{pw} \tag{6}$$

If the scans do not converge within *Num_Iterations*, then scan j is skipped and the next iteration begins. If the scans i and j converge, the translation vector $z_{d_{ij}} = \{z_{dx_{ij}}, z_{dy_{ij}}\}$ is calculated as the difference between the location of the ROV for scans i and j.

C. Scan Localization

The Scan Localization stage of the pipeline inputs the set of measured translation vectors $z_{d_{ij}}$ from the Pair Wise Scan Matching stage and aims to output the translation estimates d_{ij} for scans i and j. This is accomplished by solving a weighted least squares minimization that aims to reduce the cost function S in (7).

$$S = \sum_i \sum_{j=i-1}^{i+1} \sum_{n=1}^{4} w_{ij\_\mu\_n} (d_{ij} - \hat{d}_{ij,n})^2 \tag{7}$$

In this cost function, d_{ij} is the relative translation vector between scans i and j being estimated. The variable $\hat{d}_{ij,n}$ represents $n = 1..4$ possible measurements extracted from various combinations of d_{ij_m} as outlined below:

- $\hat{d}_{ij\_1} = z_{d_{ij}}$
- $\hat{d}_{ij\_2} = -z_{d_{ji}}$, noting that $z_{d_{ji}} \neq z_{d_{ij}}$
- $\hat{d}_{ij\_3} = z_{d_{ik}} + z_{d_{kj}}$, the reciprical pseudo measurement
- $\hat{d}_{ij\_4} = -z_{d_{jk}} - z_{d_{ki}}$ is the reciprical pseudo measurement

Each translation vector has a corresponding weight $w_{ij\_\mu\_n}$ that represents the likelihood of the measurement's accuracy at the time the particles converged. $w_{ij\_\mu\_n}$ is equal to $w_{ij\_\mu\_n}^t$ in (6) when t equals the time of convergence. The pseudo measurements' weights are calculated from the first two translation measurements' weights in the same manner.

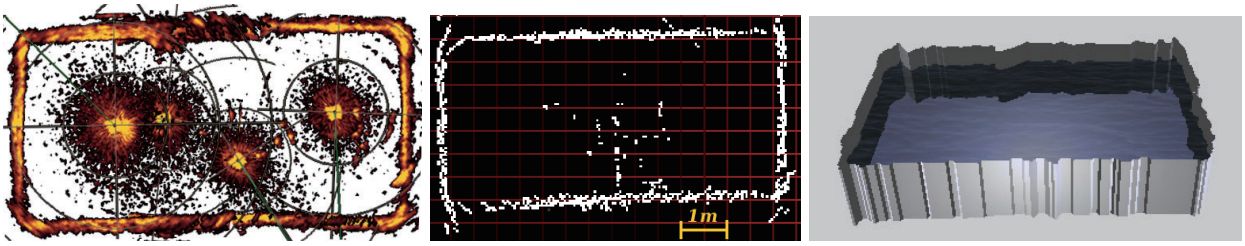
The minimizing function reduces the error among the four types of $\hat{d}_{ij,n}$.

$$min \left\{ S = \sum_i \sum_{j=i-1}^{i+1} \sum_{n=1}^{4} w_{ij\_\mu\_n} (d_{ij} - \hat{d}_{ij,n})^2 \right\}, \tag{8}$$

$$\frac{dS}{d(d_{ij})} = 0\,, \tag{9}$$

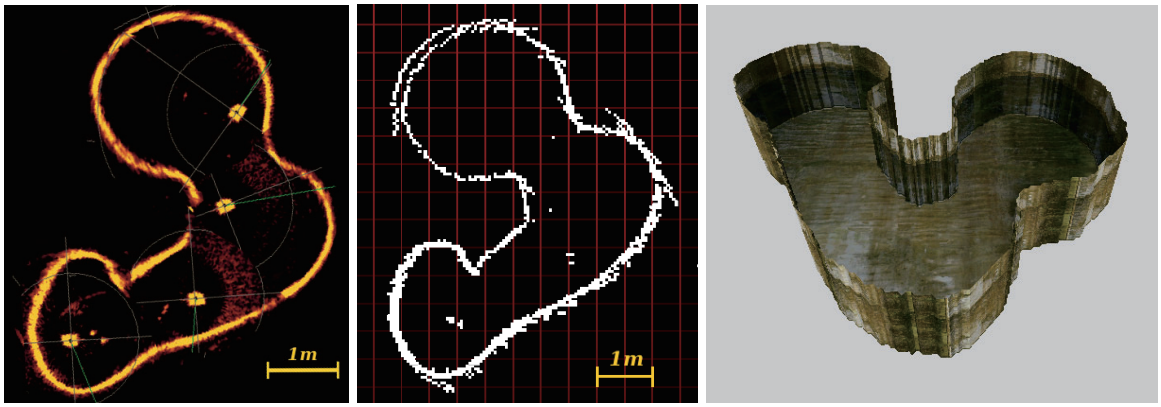
$$d_{ij} = \frac{\sum_n^4 (w_{ij\_\mu\_n} \times \hat{d}_{ij,n})}{\sum_n^4 w_{ij\_\mu\_n}} \tag{10}$$

If neither permutations of scans i and j converge then there is no possible path between scans i and j.



(a) Mosaic of Swimming Pool (b) Occupancy grid representation (c) Model from visualization

Fig. 4: Pool at Cal Poly's LAIR



(a) Cistern sonar mosaic (b) Occupancy grid from pair wise scan matching (c) Model from visualization

Fig. 5: Mdina private home cistern pipeline walkthrough

D. Map Generation

To generate a map of the robot's environment, the sonar scans are added to the robot's map with a FastSLAM algorithm that uses a log likelihood approach to combining new sonar measurements to existing cells in the robot's map. The mapping algorithm, shown in Table III, moves the robot to the localized positions according to the calculations made by the pair wise scan matching algorithm and then calls FastSLAM to update the robot's map with the new sonar scan.

In generating the map, a single reference scan is chosen to be positioned at the center of the map based on the first scan in the series that converges with another scan. The remaining scans are added using the translation vectors calculated in the *Scan Localization* step. If there is no path between scans *ref* and *i*, then the remaining scans are discarded.

The pair wise scan matching algorithm explained in this paper is currently only capable of matching scans for 2D maps. Several of the cisterns explored in Malta contained vertical walls. For these select cisterns, 3D maps were generated by extrapolating the 2D map of a single 2D slice of the cistern along the z-axis according to the depth sensor measurement.

Many of the cisterns explored were bell-shaped. To adjust for curved walls, several 2D maps of the cistern were generated each at different depths and then the walls were extrapolated along the z-axis to fill in the gaps. Manual mosaicking was used to align the scans in the third dimension.

TABLE III: Mapping Algorithm

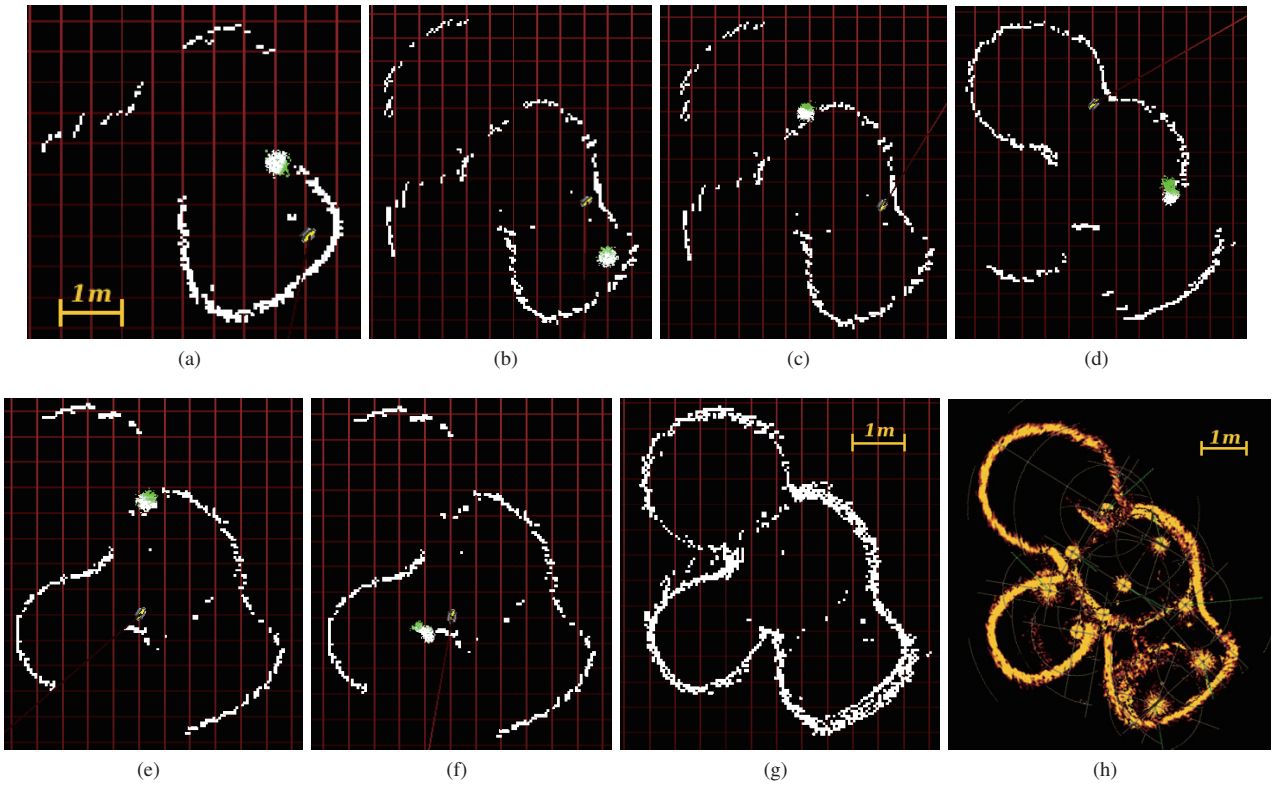
	Generate_ROV_Map(*d_{ij}*, *Num_Scans*)
1:	s_{ref} = **choose_reference_scan**(d_{ij})
2:	m_{ref} = **draw_map**(s_{ref})
3:	*for i = 1 to Num_Scans do*
4:	*if* (s_i != s_{ref})
5:	$X_{ref,i}$ = **set_ROV_state**($d_{ref,i}$)
6:	**FastSLAM**($X_{ref,i}$, s_i)
7:	*endif*
8:	*endfor*
9:	*return* m_{ref}

E. Octree-based Evidence Grid Visualization

After an octree-based evidence grid representation of the environment is generated, the map is visualized into a 3D model using isosurface extraction and then textured and visualized as described in [18].

IV. Results

The mapping algorithm presented above was applied in two scenarios. First, maps of a swimming pool located at Cal Poly's Lab for Autonomous and Intelligent Robotics (LAIR) were created. Second, the algorithm was used to create maps of ancient cisterns and tunnels found at archeological sites in Malta.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6: Pair wise scan matching alorithm on the Gatto Pardo Bistro Cistern

The swimming pool located at the LAIR was used to validate the accuracy of the mapping technique by comparing the true dimensions of the pool to the measured dimensions of the evidence grid maps. The true dimensions of the pool were obtained using a tape measure and are accurate to within 0.025m. The true width of the pool is 3.61m and the length is 7.21m. Four stationary sonar scans of the pool were obtained using the procedure for data collection explained in Sec. III-A. The mosaic of these scans can be seen in 4a. The measurements Z_m were passed into the map generation pipeline and the evidence grid with cell size 0.05m x 0.05m seen in Fig. 4b was generated.

The accuracy of the mapping algorithm was determined by comparing the actual dimensions of the pool to dimensions extracted from the 2D map. The cells inside the map were counted from one inside edge to the other. This resulted in map estimated dimensions of 3.47m in width and 7.27m in length. Since each cell is 0.05m in width and length, the pools dimensions can not be accurate to better than 0.05m. The Percent Difference (*PD*) of the ratio of the mean width and length to the ratio of the true width and length was calculated as 3.30%.

The standard deviation of the pair wise scan matching algorithm was calculated by measuring the width of the pool at 10 different locations and then comparing those 10 different measured widths and lengths to the true width and length of the pool. The calculated standard deviation of the width is 0.119m and length is 0.060m.

TABLE IV: Mean Dimension Differences (Grid vs. Mosaic)

Site Name	Width (m)	Length (m)
Mdina Home (Site 8)	0.000	-0.022
Gatto Pardo Bistro	-0.026	0.033
Swimming Pool	0.105	-0.025

This algorithm was then applied to several cisterns explored between 2008 and 2011. Results from applying the algorithm to sensor measurements taken from two different cisterns are provided here. First, a map of the Gatto Pardo Bistro was generated with a map cell size of 0.05m x 0.05m using the automated scan matching algorithm. The value used for τ_{pw} was experimentally determined as 6.0. Regarding the exponential averaging of the particle weights and their standard deviation, the following constants were used: $\chi_w = \frac{2.0}{6.0+1.0}$ and $\chi_\sigma = \frac{2.0}{4.0+1.0}$.

Fig. 6 presents the different pairings of sonar scans during the Pair Wise Scan Matching stage. The position of the ROV for the first scan used to create the map (line 2 of Table III) is represented by the yellow robot model. The localized ROV position corresponding to the second scan to be matched with the first is represented by the green to white particleswhere the whiter the color the higher the weight, (as calculated using line 4 of Table I).

Second, an interesting site explored in 2008 was found at a private home in the city of Mdina Fig. 5 (Site 8). This

cistern was mapped with the new algorithm using a cell size of 0.05m x 0.05m. This cistern demonstrates the ability to converge a set of scans where the end point scans have very limited overlap due to the robot being in two seperate chambers and very little overlap.

Since no truth data regarding the size and shape of the cistern was available, a mosaic of raw sonar scan images was manually created. The result of this manual mosaic process is shown in Fig. 5. To note, the average particle positions obtained from the automated Pair Wise Scan Matching converged to locations that correspond well to the localized ROV positions in the manually created sonar mosaic in Fig. 5. (These are the orange circles in the raw sonar scans created by acoustic reflections from the robot itself). The holes in the map are attributed to the mapping algorithm's method of fusing scans together by adding the log-odds of each cell. Modifying the merging algorithm is also a topic for future work.

Further comparison of three evidence grid-based maps was done by comparing the width and length of the mosaics to the width and length of the evidence grid. Since all the sites are not linear in shape, width and length measurements were chosen based on distinct features in the map, i.e. corners and tunnels. Table IV outlines the difference in width and length measurements between the mosaics and evidence grids of three different maps.

V. Conclusions and Future Works

A. Conclusions

This work demonstrates the ability to conduct offline mapping of underwater tunnels with robots that have low payloads. Specifically, maps were constructed without the use of robot odometry or a motion model after collecting data. By applying the newly developed pair wise scan matching algorithm, octree-based evidence grid representations of such tunnels, cisterns, wells, and water galleries were produced with a best fit solution.

The results demonstrate that for the application at hand, the algorithm is accurate up to a standard deviation of 0.119m calculated from the evidence grid of the mapped swimming pool. The percent difference between the dimensions of the pool's evidence grid map and the truth data was also measured as 3.30%. To reinforce these statistics and the accuracy of the pair wise scan matching algorithm, the dimensions of three manually created mosaic maps were compared to their corresponding evidence grid maps generated from the pair wise scan matching algorithm seen in Table IV and resulted in the highest difference being just over twice the cell size of the evidence grids.

B. Future Works

Future work includes expanding the algorithm to a sliding window approach that compares each scan to a large set of consecutive scans without introducing false positives. In the current implementation, if one of the scans does not overlap with either the scan before it or after it, then that scan along with the remaining scans are eliminated. This last limitation is avoided by collecting more scans with at least 50% overlap between consecutive scans.

Additionally, this algorithm could be expanded to support localizing an ROV in the third dimension. This is important when working in an underwater environment with nonlinear walls as do many cisterns in Malta.

References

[1] F. Loisy, P. Franois, G. Douchet, P. Hope-Darby, K. Shimmin, T. Bonner, E. Laurent, and R. Colin, "Underwater inspection experiment for a long tunnel of edf's hydroelectric facilities," in *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, oct. 2010, pp. 1 –4.

[2] P. Jasiobedzki, S. Se, M. Bondy, and R. Jakola, "Underwater 3d mapping and pose estimation for rov operations," in *OCEANS 2008*, sept. 2008, pp. 1 –6.

[3] A. Goldstein and S. Bentley, "Use of highly portable micro-sized remotely operated vehicles for environmental monitoring and mapping," in *OCEANS 2010*, sept. 2010, pp. 1 –6.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99 –110, june 2006.

[5] S. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 1793 –1798 vol.2.

[6] J. Castellanos, J. Tardos, and J. Neira, "Constraint-based mobile robot localization," in *Proceedings of the International Workshop on Advanced Robotics and Intelligent Machines (ARIM)*, 1996.

[7] E. Hernández, P. Ridao, D. Ribas, and J. Batlle, "A probabilistic scan matching algorithm using a mechanical scanned imaging sonar," *Journal on Physical Agents (JoPhA)*, vol. 3, no. 1, pp. 3–12, 2009.

[8] A. Mallios, P. Ridao, M. Carreras, and E. Hernández, "Navigating and mapping with the sparus auv in a natural and unstructured underwater environment," in *OCEANS 2011*, sept. 2011, pp. 1 –7.

[9] N. Fairfield and D. Wettergreen, "Evidence grid-based methods for 3d map matching," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 1637 –1642.

[10] A. Burguera, Y. Gonzalez, and G. Oliver, "Probabilistic sonar scan matching for robust localization," in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 3154 –3160.

[11] A. Burguera, G. Oliver, and Y. Gonzalez, "Range extraction from underwater imaging sonar data," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, sept. 2010, pp. 1 –4.

[12] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066 –1077, oct. 2008.

[13] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Fast incremental smoothing and mapping with efficient data association," in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 1670 –1677.

[14] C. White, D. Hiranandani, C. Olstad, K. Buhagiar, T. Gambin, and C. M. Clark, "The malta cistern mapping project: Underwater robot mapping and localization with ancient tunnel systems," *Journal of Field Robotics*, 2010.

[15] C. M. Clark, D. Hiranandani, C. White, M. Boardman, M. Schlachtman, P. Phillips, J. Kuehn, T. Gambin, and K. Buhagiar, "The malta cistern mapping project: Expedition ii," in *Proc. of the Unmanned Untethered Submersible Technology (UUST 09)*, 2009.

[16] C. M. Clark, C. Olstad, K. Buhagiar, and T. Gambin, "Archaeology via underwater robots: Mapping and localization within maltese cistern systems," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, dec. 2008, pp. 662 –667.

[17] S. Thrun, *Probabilistic robotics*. Cambridge, Mass: MIT Press, 2005.

[18] C. Forney, J. Forrester, B. Bagley, W. McVicker, J. White, T. Smith, J. Batryn, A. Gonzalez, J. Lehr, T. Gambin, C. M. Clark, and Z. J. Wood, "Surface reconstruction of maltese cisterns using rov sonar data for archeological study," in *Proceedings of the 7th international conference on Advances in visual computing - Volume Part I*, ser. ISVC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 461–471. [Online]. Available: http://dl.acm.org/citation.cfm?id=2045110.2045161